

# Automation Control Protocol Guide



**Version 5.2.0**

**April 6, 2017**

Last modified on May 4, 2017

## Table of Contents

Establishing and Using a Connection	5
Dataset Commands	5
audio	5
get_animating	5
get_frame_count	6
get_frame_number	6
get_frame_rate	6
get_tilt	6
kml	6
orient	7
pause	7
pic_mute	7
pip	7
pipalpha	7
pipcoords	7
pipdelay	7
pipdelete	7
pipfadein	8
pipfadeout	8
pipfps	8
pipheight	8
piphorizontal	8
pipimage	8
pipname	8
pipstyle	8
pipselect	9
piptimer	9
pipvertical	9
pipwidth	9
play	9
set_frame_rate	9
set_tilt	10
skip	10
step	10
stop	10
zrotationangle	11
zfps	11
zrotationenable	11
zrotationtoggle	11
Playlist and Library Commands	11

fadein	11
fadeout	12
get_all_name_value_pairs	12
get_clip_count	12
get_clip_info	12
get_clip_list_from_file	12
get_clip_number	13
get_playlist_name	13
load	13
next_clip	13
open_clip_list	13
open_playlist	14
overlay	14
playlist_delete	14
playlist_exists	14
playlist_read	14
playlist_rename	14
playlist_write	15
prev_clip	15
search_clip_list	15
search_clip_list_from_file	15
set_auto_presentation_mode	16
set_fadeout_duration	16
set_fadein_duration	16
<b>Interactive Commands</b>	<b>16</b>
alignment	16
annotation help	16
annotation	16
layer help	16
layer info	17
layer	17
Pointer	17
pointerimage	17
pointersize	17
splitter help	18
splitter info	18
splitter	18
zoom help	18
zoom	18
<b>System Commands</b>	<b>18</b>
bottommask	18
exit	18

get_version_number	19
get_database_date	19
get_filenames_in_directory	19
get_playlist_names_in_directory	19
get_state	19
help	19
identify	19
shutdown	19
topmask	19
Example	20

The SOS Automation Interface (also known as the show floor control protocol) is available via a TCP connection to the main SOS system. This interface allows extensive control of the operation of SOS by an external computer, as a coordinated part of a larger exhibit. For example, SOS could be synchronized with films or PowerPoint display programs, controlled by an interactive kiosk, or run on an automatic daily schedule of timed presentations.

## Establishing and Using a Connection

To use the SOS Automation Interface, establish a TCP connection to socket 2468 on the main SOS host. The SOS Automation Interface uses an ASCII line-oriented protocol. Each command is a single line of text. Every command will result in a response from the SOS system. After establishing the TCP connection, it may be kept open and used for multiple commands, but the very first command must be the word "enable". This is neither an actionable command nor a real password, but is used as a simple-minded way to avoid dealing with random socket probes and scans.

Command responses are also ASCII and line-oriented. The error response is the letter "E" (for Error) followed by a two-digit error code, all on a single line. Future error codes may be more precise, but at present this is almost always the string "E04".

The default successful command response is the letter "R" (for Ready) on a single line. This indicates successful completion of commands that have no return value (e.g. stop). It also indicates success and the end of the list for commands that return more than one line (e.g. get\_clip\_info \*). Commands that always return a single line of information (e.g. get\_frame\_count) return only that line of information in the successful case, and don't return an "R" character.

## Dataset Commands

### audio [subcommands]

The audio command controls aspects of the optional audio track associated with the current dataset loaded on the sphere. These are the available subcommands:

- **volume** - returns the current volume, for example 75.0
- **mute** - mutes the audio track.
- **unmute** - unmutes the audio track.
- **toggle** - mutes the audio track if it is playing, and turns it back on if it is muted.
- **up** - increases the volume of the audio track by a system-dependent amount, up to a maximum value.
- **down** - decreases the volume of the audio track by a system-dependent amount, down to a minimum value.
- **[0 ... 100]** - sets the volume to a percentage of the current system volume. The value is an integer in the range 0-100.

### get\_animating

The get\_animating command returns "1" if the current clip is animating, or "0" if it is paused or no clip is loaded.

### get\_frame\_count

The `get_frame_count` command returns the total number of frames in the current clip, or 0 if no clip is loaded.

### get\_frame\_number

The `get_frame_number` command returns the frame number that is currently being displayed within the current clip. Frames are numbered from 0 to one less than the total number of frames. If no clip is loaded, the frame number is zero.

### get\_frame\_rate

The `get_frame_rate` command returns the frame rate and dwell times at which the current clip is now playing. If no clip is loaded, the value is not valid or useful.

The return value is a line with three space-separated numbers: `fps first_dwell_ms last_dwell_ms`

The value `fps` is the current number of frames per second. It is a floating point number, and fractional values are allowed.

The values `first_dwell_ms` and `last_dwell_ms` give the integer dwell times for the first and last frames of the current clip, in milliseconds. They specify an additional amount of time that the system lingers on the first and last frames. This is particularly useful for animations through time, to help the viewer adjust to the time discontinuity as the clip wraps around the end.

*Example:*

```
Get_frame_rate  
30.000000 0 0
```

This example shows the default values for frame rate and dwells.

### get\_tilt

The `get_tilt` command returns the current position of the sphere display in terms of three defining Euler angles. The angles are specified in degrees. In the SOS display software, the sphere image is first rotated around the y-axis, then the x-axis, and finally the z-axis. The result is a set of three numbers that can describe any possible rotational position of the sphere.

### kml [subcommands]

The `kml` command controls the placement of KML features. By default KML legends and ancillary graphics are placed at 180 degrees longitude. These are the available subcommands:

- **legendXoffset** [dx] - offsets the longitude of the legend by dx degrees.
- **legendYoffset** [dy] - offsets the latitude of the legend by dy degrees.
- **legendoffset** [dx] [dy] - offsets the longitude of the legend by dx degrees and the latitude by dy degrees
- **pipscale** [sf] - scales all image placemarks in the KML file using sf. For example, an sf of 1.5 would increase the size by 50%.

### `orient [globe_lat,globe_lon] [room_lat,room_lon]`

The `orient` command positions the sphere display in terms of two (latitude, longitude) pairs. The intent is to provide a more intuitive way to position the sphere than the basic Euler angles specified for the `set_tilt` command. The idea is to move a particular position on the global data set (globe coordinates) to a particular position on the physical sphere (room coordinates). The global data set is further adjusted so that North is up at the specified point.

In the default SOS position (`set_tilt 0 0 0`) the two coordinate systems (globe and room) are identical ways of specifying a position on the sphere surface. As the sphere display orientation is changed, the room coordinate system by definition remains in the original position, and the globe coordinate system shifts around to describe the data orientation as displayed on the sphere.

### `pause`

The `pause` command stops animating the current clip (the opposite of `play`).

### `pic_mute`

The `pic_mute` command toggles the sos video display, turning it off if it is visible, and vice versa. When muted, the clip continues to play, but is not visible on the sphere.

### `pip [filename]`

The `pip` command creates a new PIP window and loads the specified image filename into the window. (Note that filename can also be a url image, as in `http://example.com/image.jpg`, or an rtsp stream, as in `http://server_name/stream_name.sdp`) The PIP becomes the selected PIP, to which the following PIP commands apply.

### `pipalpha [0 to 1]`

The `pipalpha` command specifies the opacity of the selected PIP. A value of 0 makes the PIP fully transparent (invisible). A value of 1 makes the PIP fully opaque. Values between 0 and 1 make the PIP translucent.

### `pipcoords [lat,lon]`

The `pipcoords` command positions the center of the selected PIP at a particular latitude and longitude. Unlike the `pipvertical` command, the PIP shape doesn't change as the latitude increases. A different transformation is used in each case, so it is recommended to use either `pipvertical` or `pipcoords`, but not both.

### `pipdelay [seconds]`

The `pipdelay` command specifies the offset in seconds from the start of the current clip where the selected PIP first appears. If not specified, the PIP is visible from the start of the clip.

### `pipdelete [name]`

The `pipdelete` command deletes the with the user-specified name. Once deleted, the PIP disappears from the sphere, and is no longer available to be manipulated by other PIP commands.

### `pipfadein [seconds]`

The `pipfadein` command specifies the number of seconds it takes the selected PIP to fade in when the PIP first appears (per the `pipdelay` command).

### `pipfadeout [seconds]`

The `pipfadeout` command specifies the number of seconds it takes the selected PIP to fade out when the PIP disappears (per the `piptimer` command).

### `pipfps [value]`

The `pipfps` command specifies the animation rate of the selected PIP in frames per second for animated pips. The default animation rate of the PIP is read from the PIP movie file. If not specified in the file, it plays at 30 fps.

### `pipheight [height]`

The `pipheight` command specifies the height of the selected PIP, in degrees latitude.

### `piphorizontal [value]`

The `piphorizontal` command specifies the horizontal placement of the selected PIP. The value is in degrees east latitude.

### `pipimage [filename]`

The `pipimage` command loads a new image file which will replace the window contents of the selected pip.

### `pipname [name]`

The `pipname` command assigns a user-specified name to the selected PIP, so that it may be selected by that name for future manipulation. If this command is not used, the name of the clip is that of the filename specified when it was created.

### `pipstyle [projector|globe|room]`

The `pipstyle` command specifies where the selected PIP appears, and how it is affected as the sphere data is reoriented via the `set_tilt` command.

- **projector** duplicates the PIP four times around the sphere, in front of the traditional projector positions. The locations are unaffected by the `set_tilt` command.
- **globe** visually attaches the PIP to the globe data set, so the PIP moves when the sphere spins or is moved with the `set_tilt` command.
- **room** keeps the PIP stationary on the sphere surface (relative to the room) as the global data set moves.



### pipselect [name]

The pipselect command selects a PIP (or multiple PIPs) with a user-specified name. The name is either specified via the pipname command, or is the filename specified originally by the pip command. Most commands then apply to the selected PIP (or PIPs).

### piptimer [seconds]

The piptimer command sets the length of time that the selected PIP remains visible on the sphere. If not specified, the PIP is always visible after it first appears.

### pipvertical [value]

The pipvertical command specifies the vertical placement of the center of the selected PIP. The value is in degrees north latitude. Shifting the PIP with this command changes its shape, since the edges of the PIP remain aligned with lines of longitude, which converge towards the north and south poles.

### pipwidth [width]

The pipwidth command specifies the width of the selected PIP, in degrees longitude.

### play [clip\_number]

The play command with the clip\_number specified clears the sphere, then loads and plays the given clip\_number. The first clip in the playlist is clip\_number 1. For example, play 2 loads and plays the second clip in the current playlist. A clip number that is greater than the last clip in the current playlist plays the last clip.

The play command with no arguments does one of two things, depending on whether or not a clip is currently loaded. If the sphere is clear (nothing loaded), play loads and plays the first clip in the current playlist (identical to play 1). If a clip is already loaded, play starts animating the clip (the opposite of pause). If the clip is already animating, play has no effect.

### set\_frame\_rate [fps] ([first\_dwell\_ms last\_dwell\_ms])

The set\_frame\_rate command sets the playback speed of the currently loaded clip. The fps argument is required, and specifies the speed of animation in frames per second. If not specified, the default animation rate is 30 frames per second. The fps value is a floating point number, and fractional values are allowed.

The optional arguments first\_dwell\_ms and last\_dwell\_ms specify the dwell times for the first and last frames of the current clip, in milliseconds. They specify an additional amount of time to linger on the first and last frames. This is particularly useful for animations through time, to help the viewer adjust to the time discontinuity as the clip wraps around the end.

Note: The SOS system does not attempt to impose a maximum frame rate, because it depends on the part of the computer and the clip being played. Most clips will eventually fail to play correctly as the frame rate is increased, due to exhaustion of computer system resources.

#### *Example:*

set\_frame\_rate 29.97 sets the frame rate to 29.97 frames per second (the historic American TV standard frame rate).

*Example:*

`set_frame_rate 30 1000 500` sets the frame rate to 30 frames per second (the SOS default), and dwells an additional second on the first frame, and half-second on the last frame.

### `set_tilt [x] [y] [z]`

The `set_tilt` command positions the sphere display in terms of three defining Euler angles. The angles are specified in degrees. In the SOS display software, the sphere image is first rotated around the y-axis, then the x-axis, and finally the z-axis. The result is a set of three numbers that can describe any possible rotational position of the sphere.

### `skip [frame_number]`

The `skip` command stops animation, then jumps to the specified absolute frame number within the current clip. The first frame of the clip is number 0. Values of `frame_number` less than zero are rounded up to zero. Values of `frame_number` past the last frame are rounded down to the last frame.

*Example:*

`skip 0`

Jumps to the beginning of the current clip.

*Example:*

`skip 99`

If there are 100 frames in the current clip, this jumps to the last frame of the clip.

*Example:*

`skip 999999999`

Jumps to the end of the current clip. 999999999 is larger than any reasonably-sized clip, and is rounded down to the last frame.

### `step [n_frames]`

The `step` command stops animation, then increments the current frame number by `n_frames`, and jumps to the new frame. If `n_frames` is negative, the command steps backward. The command argument is the number of frames to jump relative to the current frame number. The `step` command wraps smoothly around the first or last frame of the clip as needed.

*Example:*

`step 10`

If there are 100 frames loaded, and SOS is on frame number 10, this command jumps to frame number 20.

*Example:*

`step 1`

If there are 3600 frames loaded, and SOS is paused on frame number 3599 (the last frame in the clip), this command jumps to frame 0 (the first frame in the clip).

### `stop`

The `stop` command clears the sphere. It unloads any data that is currently loaded.

### zrotationangle [angle]

The zrotationangle command sets the increment angle for Z Rotation, in degrees. Every time the rotation angle is updated (at a rate determined by zfps), it is incremented by this amount. The default is 0.1 degrees.

*Example:*

```
zfps 1
```

```
zrotationangle 6
```

```
zrotationenable 1
```

This set of commands make the sphere act like a jumpy second hand on a watch, jumping by 6 degrees every second, so the sphere makes one complete rotation around the room every minute.

*Example:*

```
zfps 30
```

```
zrotationangle 0.1
```

```
zrotationenable 1
```

This set of commands turn on Z rotation with the same settings as the default value. Since the rotation angle is being updated 30 times per second, the result is a smooth rotation around the room. With a rotation angle of 0.1 degree being incremented 30 times per second, the sphere rotates 3 degrees every second, so it completely rotates around the room every 2 minutes (since  $360 \text{ degrees} / 3 \text{ degrees per second} = 120 \text{ seconds}$ ).

### zfps [fps]

The zfps command sets the number of times per second that the rotation angle is updated. The default value is 30.

### zrotationenable [0 | 1]

The zrotationenable command turns Z Rotation on or off. A command argument of 0 or 1 is required: 1 to turn Z Rotation on and 0 to turn it off.

### zrotationtoggle

The zrotationtoggle command toggles the status of Z Rotation, turning it on if it is off, and vice versa.

## Playlist and Library Commands

While most SOS automation commands apply to the current clip, those in this section apply to the playlist as a whole. An SOS playlist is a set of clips, numbered from one through the number of clips in the playlist.

### fadein [fade\_duration]

The fadein command fades the sphere in from black. The fade\_duration parameter specifies the number of seconds over which to fade in from black. If the fade\_duration is not specified, the default fadein duration is used. This command is called automatically after a clip is loaded.

### fadeout [fade\_duration]

The fadeout command fades the sphere to black. The fade\_duration parameter specifies the number of seconds over which to fade to black. If the fade\_duration is not specified, the default fadeout duration is used. This command is called automatically before a clip is unloaded.

### get\_all\_name\_value\_pairs [clip\_number]

The get\_all\_name\_value\_pairs command returns a list of name-value pairs for all the parameters specified for the given clip\_number in the currently loaded playlist. The syntax of the returned list is derived from the Tcl scripting language, and is a single line containing a set of space delimited name-value pairs. Each name-value pair is also space delimited. If the value contains internal spaces, it is enclosed in curly braces. If the value itself contains curly braces, they and other special characters may be further escaped with backward slashes.

Once a clip is loaded, many of the clip attributes can be modified by the automation interface. The get\_all\_name\_value\_pairs command returns the initial values of those attributes as recorded in the playlist, even if they have changed since the clip was loaded.

### get\_clip\_count

The get\_clip\_count command returns the total number of clips in the currently loaded playlist.

### get\_clip\_info [clip\_number | \* clip\_parameter]

The get\_clip\_info command returns information about a clip as specified within the currently loaded playlist.

The first command argument is required, and is the clip number, or an asterisk for all clips.

The second command argument is optional, and is the name of the clip parameter as specified in the playlist. The default parameter is the name of the clip.

Once a clip is loaded, many of the clip attributes can be modified by the automation interface. The get\_clip\_info command returns the initial values of those attributes as recorded in the playlist, even if they have changed since the clip was loaded.

The value returned by the get\_clip\_info command has a different format depending on whether a specific clip number was requested, or the asterisk was used to request all clips. If a specific clip number was given, the return value is one line, with the value of the requested parameter (or an empty line if the parameter was not specified for this clip). If the asterisk was used to request all clips, the return value is a set of lines, one per clip, with a final line consisting of the character "R" to indicate the end of the list. The line for each clip in this case is the clip number followed by a space character, optionally followed by the parameter value for that clip (if specified in the playlist).

### get\_clip\_list\_from\_file [filename]

The get\_clip\_list\_from\_file command returns an unsorted list of all the name-value pairs of each clip found in the filename, where filename is the absolute path to a .sos playlist file. The syntax of the returned list is derived from the Tcl scripting language, and is a single line containing a set of space delimited name-value pairs (see the get\_all\_name\_value\_pairs command for more information on the Tcl syntax).

Each name tag in a name-value pair is prepended by the clip's sequential number in the playlist file (for

example: 2,category, where category is the name tag).

Note that there are two extra name-value pairs always included in the return list:

0,animate value (where value is 0 or 1)

size value (where value is the number of clips in the playlist)

In addition, there is one extra name-value pair for each clip that is always included in the return list, but is not present in the clip's .sos file:

#,valid value (where # is a clip number in the playlist, and value is 0 or 1)

### *Example:*

Assume I have a file called myplaylist.sos located in the directory ~/sosrc/. myplaylist.sos points to two clips, Blue Marble (23 degree tilt) and Helium Sun. If get\_clip\_list\_from\_file "~/sosrc/myplaylist.sos" is called, the following is returned:

```
1,opengl true 1,publisher NOAA size 2 0,animate 0 1,category land 2,publisher {NASA GSFC}
1,fps 40 2,fps 30 2,name {Helium Sun} 2,pipcount 0 1,tiltx 23.5 2,category astronomy 0,valid 1
1,valid 1 1,creator NASA 2,creator {NASA GSFC} 2,valid 1 1,name {Blue Marble (23 degree tilt)}
1,datadir /shared/sos/media/land/blue_marble/blue_marble/4096.jpg 2,datadir
/shared/sos/media/astronomy/helium_sun/helium_sun_2048.mp4 1,pipcount 0
```

## get\_clip\_number

The get\_clip\_number command returns the number of the currently playing clip. Clips in the playlist are numbered from one to the total number of clips in the playlist.

If the currently playing clip is not part of the current playlist, get\_clip\_number returns 0. This happens when a new playlist is loaded while a clip continues to play.

If no clip is currently playing, get\_clip\_number returns -1.

## get\_playlist\_name

The get\_playlist\_name command returns the filename of the current presentation playlist. If no presentation playlist is currently loaded, it returns an empty line.

## load [filename | dirname]

The load command loads an individual global image file (or directory of files) without the context of a playlist. Ordinarily, data sets are viewed as defined by a clip within a playlist, but this command allows loading an image file without editing a playlist. A temporary clip, number 0, is created when this command is used.

## next\_clip

The next\_clip command stops any currently playing clip, clears the sphere, and plays the next clip sequentially in the playlist. If the last clip was playing, it jumps to the first clip.

## open\_clip\_list [clip\_list]

The open\_clip\_list command clears the currently loaded playlist and loads and opens a new temporary playlist in SOS Stream GUI containing all the clips in clip\_list. clip\_list is a Tcl list of name-value pairs for a

set of clips (for example, a list that is in the format returned by the `get_clip_list_from_file` command).

### `open_playlist [playlist_file]`

The `open_playlist` command loads a new playlist file. If the `playlist_file` name starts with a forward slash character, it specifies an absolute path name. Otherwise, it specifies a file relative to `~/sosrc` (the `sosrc` directory in the user's home directory). Any currently playing clip continues to play after a new playlist is loaded.

*Example:*

```
open_playlist /shared/sos/media/land/blue_marble/blue_marble/playlist.sos
```

This example opens the library playlist for the blue marble clip, within the SOS media tree.

```
open_playlist normal-demo.sos
```

This example opens the default SOS playlist `~/sosrc/normal-demo.sos`.

### `overlay [filename | dirname | clip_number]`

The `overlay` command creates a new layer on top of any existing global data, and loads the specified global image file or directory of files to the new layer. The new layer is created with default attributes which can then be changed with the `layer` command. When an integer `clip_number` is specified, the global image layers of that clip are loaded as new layers on top of any existing global data.

### `playlist_delete`

`[playlist_file]`

The `playlist_delete` command deletes the specified presentation playlist file, which is required to use a relative path under the presentation playlist directory `~/sosrc`.

*Example:*

```
playlist_delete mysubdir/playlist.sos
```

### `playlist_exists`

`[playlist_file]`

The `playlist_exists` command checks to see if the specified playlist file exists. The playlist is required to use a relative path under the presentation playlist directory `~/sosrc` or a full path under a NOAA dataset directory (i.e., under `/shared/sos/media` or `/shared/sos/rt/noaa`).

*Example:*

```
playlist_exists /shared/sos/media/site-custom/mysubdir/myplaylist.sos
```

### `playlist_read [playlist_file]`

The `playlist_read` command opens and reads the contents of the specified playlist file, which is returned as a string. The playlist is required to use a relative path under the presentation playlist directory `~/sosrc` or a full path under the NOAA dataset directory (i.e., under `/shared/sos/media` or `/shared/sos/rt/noaa`).

*Example:*

```
playlist_read /shared/sos/media/site-custom/mysubdir/myplaylist.sos
```

### `playlist_rename`

`[from_playlist_file] [to_playlist_file]`

The `playlist_rename` command renames (i.e., moves) the specified playlist file to a new name. Both the “from” and “to” playlist\_files are required to use a relative path under the presentation playlist directory `~/sosrc` and the “to” playlist\_file cannot be for an existing file. Existing subdirectories within `~/sosrc` may be specified, but new subdirectories are not created by this command.

Example: `playlist_rename myplaylist.sos mysubdir/myplaylist2.sos`

### playlist\_write

[playlist\_file] [playlist\_text]

The `playlist_write` command writes a playlist-formatted text string into the specified playlist file, creating it, if necessary. Ends of lines in the text are indicated by “\n”. The playlist is required to use a relative path under the presentation playlist directory `~/sosrc`. Existing subdirectories within `~/sosrc` may be specified, but new subdirectories are not created by this command.

Example: `playlist_write myplaylist.sos “# comment\ninclude=/shared/sos/media/site-custom/playlist.sos\n#> presenter note\n”`

### prev\_clip

The `prev_clip` command stops any currently playing clip, clears the sphere, and plays the previous clip sequentially in the playlist. If the first clip was playing, it jumps to the last clip.

### search\_clip\_list [search\_term] [clip\_list list\_of\_tags]

The `search_clip_list` command is identical to the `search_clip_list_from_file` command (see below). The only difference is the second parameter, `clip_list`. `clip_list` is a Tcl list of name-value pairs for a set of clips (for example, a list that is in the format returned by the `get_clip_list_from_file` command).

### search\_clip\_list\_from\_file [search\_term] [playlist\_file] [list\_of\_tags]

The `search_clip_list_from_file` command searches through the name-value pairs of each clip in the `playlist_file` and returns all of the name-value pairs of only those clips where each of the search terms were present somewhere in the clip's values.

The first command argument is the term or terms to be searched for. Multiple search terms should be enclosed in quotation marks (ex. “carbon aerosol”).

The second command argument is the absolute path to a .sos playlist file.

The third command argument, `list_of_tags`, selects a specific name tag to search in the clip's sos playlist file. There are three options:

- `name` (search only the values paired with the name tag)
- `keywords` (search only the values paired with the keywords tag)
- `""` (the empty string means search all tags specified for the clip)

Multiple tags may be specified by enclosing them in quotation marks (ex. “name keywords”). Note that at least one of the above options must be specified.

In the return list of search results, each name tag in a name-value pair is prepended by the clip's number in the playlist file (for example: 2,category, where category is the name tag).

Note that there is one extra name-value pair for each clip that is always included in the list that is not found

in the clip's .sos file:

0,valid value (where value is 0 or 1)

*Example:*

```
search_clip_list_from_file "helium sun" "~/sosrc/___library___/all.sos" name
```

```
196,publisher {NASA GSFC} 196,datadir
```

```
/shared/sos/media/astronomy/helium_sun/helium_sun_2048.mp4 196,valid 1 196,name {Helium  
Sun} 196,pipcount 0 196,category astronomy 196,fps 30 196,creator {NASA GSFC}
```

### set\_auto\_presentation\_mode [0 | 1]

The set\_auto\_presentation\_mode turns on or off (0 = off, 1 = on) the auto presentation (or auto run) mode of SOS. While in auto presentation mode, the system automatically advances from one clip to the next. The system plays each clip the number of seconds that are specified by the timer value for that clip in the current playlist. If no timer value is specified for the clip, a system default time is used (typically 2 or 3 minutes per clip).

### set\_fadeout\_duration [seconds]

The set\_fadeout\_duration command sets the default duration in seconds for subsequent fadeouts.

### set\_fadein\_duration [seconds]

The set\_fadein\_duration command sets the default duration in seconds for subsequent fadeins.

## Interactive Commands

### alignment [alignment\_subcommand]

Note: The alignment subcommands are reserved for use by SOS software to align the sphere. It is not recommended that third-party software make use of these commands, as they may change with any future SOS software release. They are documented here for informational purposes only.

### annotation help

This command describes the annotation command options

### annotation [subcommand]

The annotation command is used for interactive drawing on the sphere. The subcommand is a list of one or more of the following:

- **clear** - clears the annotation
- **drawline** [lat1,lon1,lat2,lon2,width,red,green,blue] - draw a line based on the parameters that are provided for position, width and color.

### layer help

This command describes the layer command options



### layer info

This command gives you information about layers in the currently loaded dataset, including PIPs and labels.

### layer [index] [subcommands]

This command lets you modify the existing layers. The index value can be top (the top layer), pip or label, or an integer layer index (zero-based). You can get this information by first issuing the "layer info" command.

The subcommands available are:

- **[name]** - renames the layer to the provided name
- **on** - turns the layer off
- **off** - turns the layer on
- **alpha** [0.0-1.0] - adjusts the transparency
- **minzoom** [float] - adjusts the minimum zoom level
- **maxzoom** [float] - adjusts the maximum zoom level
- **east** [float] - sets the east boundary of the layer
- **west** [float] - sets the west boundary of the layer
- **north** [float] - sets the north boundary of the layer
- **south** [float] - sets the south boundary of the layer
- **delete** - deletes the layer

### Pointer [subcommand]

The pointer command takes multiple arguments for manipulating the state of the associate sphere image.

- **on** - turns on the pointer, making it visible on the sphere.
- **off** - turns off the pointer, making it disappear from the sphere.
- **red** - makes the pointer image a red simulated laser pointer.
- **green** - makes the pointer image a green simulated laser pointer.
- **grab** - makes the sphere "stick" to the pointer, so the sphere data set moves as the pointer is moved.
- **ungrab** - releases the sphere after a pointer grab, so the sphere no longer moves with the pointer.
- **[lat,lon]** - moves the pointer to an absolute position specified by latitude,longitude. No white space is allowed in lat,lon.
- **[Dlat,lon]** - moves the pointer by an incremental "delta" latitude,longitude. No white space is allowed in Dlat,lon.

### pointerimage [filename]

The pointerimage command gives the name of an image file to use as the pointer image. This allows other types of pointers than the default red or green simulated laser pointer. This should be a png file with a transparent background.

### pointersize [degrees]

The pointersize command specifies the diameter of the simulated laser pointer image. The argument is the width and height of the underlying image, in degrees at the equator.

### splitter help

This command describes the splitter command options

### splitter info

This command lists all the information about the current state of the splitter

### splitter [subcommands]

The splitter command controls an interactive feature to split the sphere into 2-4 slices and replicate the section of data currently being displayed on the sphere at the center of the User Position (the red dot by default, or latlon below). The subcommands that are available are:

- **on** - turns the splitter on
- **off** - turns the splitter off
- **latlon** [float,float] - defines where the splitter should be centered
- **slices** [int] - sets the number of slices
- **write** - displays the splitter with the current settings

### zoom help

This command describes the zoom command options

### zoom [subcommand]

The zoom command sets and displays attributes of the virtual magnifying glass, which is used to zoom in on global sphere data. The subcommands that are available are:

- **on** - turns the zoom on
- **off** - turns the zoom off
- **factor** [float] - defines the level of zoom
- **winsize** [float] - sets the size of the zoom window
- **latlon** [float,float] - set the position of the zoom window

## System Commands

### bottommask [on | off | lolat,hilat]

The bottommask command turns the bottom sphere mask on or off, or specifies the latitude range over which the masking transitions from no masking to fully masked. It is used to provide a faded transition to the "black hole" at the bottom of the sphere where the projectors are not able to illuminate the sphere. It can help reduce rough edges and moire patterns, at the cost of decreasing the overall size of the illuminated sphere area.

### exit

The exit command closes the current client connection to the SOS system.

### get\_version\_number

The get\_version\_number command returns a string identifying the running version of SOS.

### get\_database\_date

The get\_database\_date command returns a string containing the timestamp of the SQLite database containing the SOS Data Catalog (the file path is /shared/sos/database/sos\_sqlite.db). An example of the timestamp format is 2015-03-05 17:27:47.087217857 -0700, which corresponds to March 5, 2015, slightly after 5:27:47 PM in the Mountain Daylight Time zone (7 hours before GMT).

### get\_filenames\_in\_directory [directory\_name]

The get\_filenames\_in\_directory command returns a list of the files in the specified directory on the SOS system.

### get\_playlist\_names\_in\_directory [directory\_name]

The get\_playlist\_names\_in\_directory command returns a list of the playlist files in the specified directory on the SOS system. If directory\_name is not specified, the default playlist directory is used (~/.sosrc). Note: Use of the directory\_name argument is currently not supported.

### get\_state [timestamp]

This command returns a list of the SOS state variables that have changed since the specified timestamp. It is intended to provide a simple and efficient polled interface for updating a user interface to SOS. It returns a list of {name value} pairs. Each pair is enclosed in curly braces, and the individual pairs are delimited by spaces. One of the pairs is a timestamp that can be used for future calls.

To use get\_state in a polling loop, first call it with a timestamp value of 0. It returns the current value for all the available state variables, including the timestamp. Then when the returned time stamp is used in a future call, only values that have changed are returned (including a new timestamp).

### help

The help command returns a brief summary of all the automation control interface commands.

### identify [on | off]

Note: This command is very likely to be discontinued.

The identify command displays information on the sphere to identify the individual projectors by name: P1, P2, P3, or P4.

### shutdown

The shutdown command shuts down the running SOS software on the system.

### topmask [on | off | lolat,hilat]

The topmask command turns the top sphere mask on or off, or specifies the latitude range over which the masking transitions from no masking to fully masked. It is used to provide a faded transition to the "black hole" at the top of the sphere where the projectors are not able to illuminate the sphere. It can help reduce rough edges and moire patterns, at the cost of decreasing the overall size of the illuminated sphere area.

## Example

Here is a trivial example of the automation interface, using telnet to establish an interactive session. Commands typed by the user are shown in **bold**.

```
sos@thing1:~$ telnet localhost 2468
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
enable
R

get_clip_info *
1 Blue Marble - Hi-Res
2 Hurricane Season - 2005
3 Sea Surface Currents
4 Sea Surface Temperature NOAA Model (black land)
5 Sun: X-Ray - 2003
6 Mars
7 Tsunami Wave Propagation: Indian Ocean - December 26, 2004
8 Aerosols: Black Carbon and Sulfate
9 Climate Model: Temperature Change (GFDL a1b) - 1870 - 2100
10 Nighttime Lights
11 Air Traffic
12 NOAA Logo
13 Fossil Fuel: CO2 Release - 2011-2012
14 Surface Temperature
R
get_clip_number
1
get_frame_count
3600
get_frame_number
179
get_frame_number
504
exit
Connection closed by foreign host.
sos@thing1:~$
```